

# JS额外知识补充

王红元 coderwhy

# with语句

- **with语句** 扩展一个语句的作用域链。

```
var obj = {  
  name: "hello world",  
  age: 18  
}  
  
with(obj) {  
  console.log(name)  
  console.log(age)  
}
```

- **不建议**使用with语句，因为它可能是混淆错误和兼容性问题的根源。

# eval函数

- eval是一个特殊的函数，它可以将传入的字符串当做JavaScript代码来运行。

```
var evalString = `var message = "Hello World";console.log(message)`  
eval(evalString)  
  
console.log(message)
```

- 不建议在开发中使用eval：

- eval代码的可读性非常的差（代码的可读性是高质量代码的重要原则）；
- eval是一个字符串，那么有可能在执行的过程中被刻意篡改，那么可能会造成被攻击的风险；
- eval的执行必须经过JS解释器，不能被JS引擎优化；

# 认识严格模式

- 在ECMAScript5标准中，JavaScript提出了**严格模式的概念（Strict Mode）**：
  - 严格模式很好理解，是一种**具有限制性的JavaScript模式**，从而使**代码隐式的脱离了“懒散（sloppy）模式”**；
  - **支持严格模式的浏览器**在检测到代码中有严格模式时，会**以更加严格的方式对代码进行检测和执行**；
- 严格模式对正常的JavaScript语义进行了一些限制：
  - 严格模式通过 **抛出错误** 来消除一些原有的 **静默（silent）** 错误；
  - 严格模式让**JS引擎在执行代码时可以进行更多的优化**（不需要对一些特殊的语法进行处理）；
  - 严格模式禁用了**在ECMAScript未来版本中可能会定义的一些语法**；

# 开启严格模式

- 那么如何开启严格模式呢？严格模式支持粒度话的迁移：
  - 可以支持在js文件中开启严格模式；
  - 也支持对某一个函数开启严格模式；
- 严格模式通过在文件或者函数开头使用 `use strict` 来开启。

```
// "use strict";  
  
// 使用let作为标识符的名称  
var name = "abc"  
console.log(name)  
  
// 定义变量时不使用var  
var message = "Hello World"  
console.log(message)
```

```
function foo() {  
  "use strict";  
  
  m = "foo"  
  console.log(m)  
}  
  
foo()
```

# 严格模式限制

■ 这里我们来说几个严格模式下的严格语法限制：

- JavaScript被设计为新手开发者更容易上手，所以有时候本来错误语法，被认为也是可以正常被解析的；
- 但是这种方式可能给带来留下来安全隐患；
- 在严格模式下，这种失误就会被当做错误，以便可以快速的发现和修正；

■ 1. 无法意外的创建全局变量

■ 2. 严格模式会使引起静默失败(silently fail,注:不报错也没有任何效果)的赋值操作抛出异常

■ 3. 严格模式下试图删除不可删除的属性

■ 4. 严格模式不允许函数参数有相同的名称

■ 5. 不允许0的八进制语法

■ 6. 在严格模式下，不允许使用with

■ 7. 在严格模式下，eval不再为上层引用变量

■ 8. 严格模式下，this绑定不会默认转成对象